Measurement of a Scientific Workload using the IBM Hardware Performance Monitor

Robert J. Bergeron
bergeron@nas.nasa.gov

NAS Systems Division
NASA Ames Research Center
Mail Stop 258-5
Moffett Field, CA 94035-1000

Abstract

This paper presents data on the performance of the NAS SP2 IBM system using RS6000 hardware monitors and a performance measurement tool. The data collected showed that the SP2 averages about 1.3 Gflops, about 3% of peak. The report provides the relative usage for the various hardware units over the entire workload measured over a 9-month period. The workload displays moderate parallelism, with the most popular choice of nodes as 16. Although the monitor data provide a good snapshot of workload performance, causal correlations regarding key performance indicators appear difficult to draw from the current data due to the absence of I/O delay measurements.

1. Introduction

For many years, supercomputers have employed hardware monitors designed and built into their custom processors to display individual user code characteristics and total system performance.Recently, RISC processors have begun to supply hardware monitors which are software accessible to users.
In the 1995-1996 period, NAS used a combination of RS6000 hardware monitors and a performance measurement tool to monitor the performance of its floating-point, memory intensive workload on a cluster of IBM RISC POWER2 processors. This monitor and the software which provides an interface to the monitor counters permit a detailed description of the CPU instructions executed, counts and delays associated with cache and TLB misses, and utilization of the various execution elements. This paper will describe the computer system used to execute the workload, the monitor used to report performance characteristics, and some general characteristics of this workload. Then the paper will describe the workload measurements, the characteristics of the batch jobs, and the degree of parallelism in this workload. Finally, the paper will provide some remarks regarding the performance monitoring of parallel systems.

2.Description of the System

The NAS SP2 is a distributed memory 144-node RS6000/590 cluster connected by a high-performance proprietary IBM network. The RS6000 nodes comprising the cluster consist of semi-custom chips or units described briefly as follows [White and Dhawan,1994]:

o-The Instruction Decode Unit(ICU)
This unit prefetches instructions from the instruction cache. The ICU executes branch and condition register instructions and dispatches instructions to the Fixed Point Units and the Floating Point Units. The ICU can prefetch 8 instructions/cycle from the instruction cache and can dispatch 4 instructions/

cycle to the Fixed Point Units and the Floating Point Unit

o-The Fixed Point Unit(FXU)
This dual unit (FXU0 and FXU1) processes all storage references, integer arithmetic, and logical operations. The FXU can execute 2 instructions (including 2 storage references) per cycle. The unit generates addresses in its general purpose registers(GPRs). The FXU also performs data cache directory searches.

o-The Floating Point Unit(FPU)
This dual unit (FPU0 and FPU1) contains Floating Point Registers(FPRs) and 2 64-bit execution units. When executing a compound floating point multiply add instruction (fma), the FPU can produce 4 floating point operations(FLOPS) per cycle. The FPU also contains hardware to overlap floating point stores with its arithmetic operations.

o-The Data Cache Unit(DCU)
The DCU supports two one-word data busses to the FXU, two quad-word busses to the FPU, a four-word instruction bus to the ICU and a two-word system I/O (SIO) bus to the I/O subsystem for Direct Memory Access(DMA) support. This cache is a four-way set-associative dual ported cache consisting of four identical chips. RISC processors employ a memory hierarchy to transport data from the off-chip memory to the cache and then to the registers.

o-The Storage Control Unit(SCU)
This unit controls communications between the CPU, the memory and the SIO bus.

o-The I/O Unit(SCU)
This unit controls the I/O by implementing a 64-bit streaming data protocol on the Power2 data line, the Micro Channel.

Each node has at least 128 Mbytes of main memory and 2 Gbytes of disk space. The NAS SP2 processors provide a 4-way set associative data cache of 256 kB, arranged in 1024 lines of 256 bytes each. The IBM RISC 6000 implements virtual memory with a page size of 4096 bytes and supports 512 entries in the TLB. Each of the 144 nodes executes a copy of the IBM version of UNIX. For the period monitored, this version was AIX 4.1.3. The SP2 processor operates at a clock rate of 66.7 Mhz, and displays a peak performance of 267 Mflops.
The nodes were interconnected by the High Performance Switch (Stunkel et al, 1995) through communication adapters attached to the node input/output bus. This network displayed a latency of approximately 45 microseconds and a bandwidth of 34 Mbyte/second. The available communication bandwidth over this switch scales linearly with the number of processors. Extensive testing at NAS indicated this switch allowed a variety of parallel applications to scale well and the system displayed little performance degradation when tested under a full load of message-passing jobs.
The NAS SP2 provided an NFS-mounted external filesystem accessible by all nodes with 3 home filesystems of 8 GB each. Data transfers from the SP2 nodes to the home filesystems also occurred over the switch.
NAS employed its Portable Batch System(PBS) for job management. Key features of PBS included support for parallel job scheduling and direct enforcement of resource allocation policies. PBS also provides interactive login to the SP2 nodes, which allows users to more easily debug message-passing programs.

3. Description of the Monitor

The SP2 POWER2 Performance Monitor consist of 22 32-bit counters located on the SCU chip which can report CPU and storage-related events. The POWER2 counters provide a set of 5 counters and 16 reportable events each for the FPU, the FXU, the ICU, and the SCU. The selected 22 events are a subset of the 320(some overlapping) signals which can be selected and reported by software [Welbon,1994].

-NAS Counter Selection

The hardware monitor allows many possible combinations of events, but each combination must be implemented and verified in the monitoring software.The NAS counter selections, shown in Table 1, were chosen to give a broad overview of workload CPU performance.

Table 1: NAS SP2 RS2HPM Counters

Counter	Label	Description
user.fxu0	FXU[0]	number of instructions executed by Execution unit 0
user.fxu1	FXU[1]	number of instructions executed by Execution unit 1
user.dcache_mis	FXU[2]	FPU and FXU requests for data not in the D-cache
user.tlb_mis	FXU[3]	FPU and FXU requests for data not in the D-cache
user.cycles	FXU[4]	user cycles
user.fpu0	FPU0[0]	arithmetic instructions executed by Math 0
fpop.fp_add	FPU0[1]	floating point adds executed by Math 0
fpop.fp_mul	FPU0[2]	floating point multiplies executed by Math 0
fpop.fp_div	FPU0[3]	floating point divides executed by Math 0
fpop.fp_muladd	FPU0[4]	floating point multiply-adds executed by Math 0
user.fpu1	FPU1[0]	arithmetic instructions executed by Math 1
fpop.fp_add	FPU1[1]	floating point adds executed by Math 1
fpop.fp_mul	FPU1[2]	floating point multiplies executed by Math 1
fpop.fp_div	FPU1[3]	floating point divides executed by Math 1
fpop.fp_muladd	FPU1[4]	floating point multiply-adds executed by Math 1
user.icu0	ICU[0]	number of type I instructions executed
user.icu1	ICU[1]	number of type II instructions executed
user.icache_reload	SCU[0]	data transfers from memory to the I-cache
user.dcache_reload	SCU[1]	data transfers from memory to the D-cache
user.dcache_store	SCU[2]	number of transfers of D-cache data to memory
		occurs when the D-cache destination for incoming data currently contains data which has been modified
user.dma_read	SCU[3]	data transfers from memory to an I/O device
user.dma_write	SCU[4]	data transfers to memory from an I/O device

The monitor reports floating-point adds, multiplies, and fma operations. The fma operation counts as an add and a multiply for the purpose of flop counting. An implementation error in the hardware monitor prevented the proper reporting of the division operations, which typically constitute about 3% of the total floating operations for the NAS workloads. The monitor also reports the number of instructions issued by the floating-point units. The instructions issued by the fixed point units are predominantly memory loads and stores for the CFD codes in the NAS workload.The monitor also reports instruction and data cache reloads and misses.The Direct Memory Access (DMA) counters report the level of I/O activity for data moved between memory and the I/O devices; these counters also report the amount of I/O associated with message-passing.

IBM did not distribute software to access the RS6000 hardware counters, but in 1995 Jussi Maki of the Center for Scientific Computing in Finland created a set of tools for monitoring the POWER2 hardware counters [Maki,1995]. These tools allowed the reporting of events occurring in both user and system mode thru a multipass sampling mode. After NAS had obtained IBM's approval, Bill Saphir(NERSC) installed these tools on the SP2. These tools, collectively termed RS2HPM herein, consist of library, data collection daemon, kernel extension and other utilities. Bill Saphir also introduced some valuable extensions of these tools to allow monitoring of individual job performance, as well as global system performance [Saphir,1996].

-System-wide data collection

The RS2HPM daemon, executing on all nodes of the SP2, allows automatic sampling and data access over the network via TCP. At 15-minutes intervals, the cron daemon runs a script to collect data from all the SP2 nodes which are available for user jobs and stores this data for later analysis. This daemon collects performance data from the nodes whether or not user processes are executing.

-Batch job data collection

The PBS batch system runs a prologue script before each job and an epilogue script after each job. These scripts know which SP2 nodes the batch job is using and obtain counter values at the beginning and end of each job for these nodes. These values are written to a file for later processing and viewing by both users and system personnel. For individual programs to be reported, users must place commands into their batch scripts or preface interactive sessions with the appropriate RS2HPM commands.

4. Workload Description

The NAS workload consists primarily of codes solving computational fluid dynamics problems involving aerodynamics, hypersonics, propulsion, and turbulence. Many of the aerodynamics workload codes perform parallel multidisciplinary optimization which involves systematically modifying an aircraft configuration to maximize or minimize a chosen aerodynamic figure of merit. This approach involves coupling a CFD solver to a numerical optimization procedure and should display a high degree of parallelism since computations on the various configurations are completely independent.

Other aerodynamics codes, constituting the majority of the NAS SP2 workload, involve multiple grids treating a single aircraft. There are a variety of numerical methods for treating such problems, but most would involve the following steps. The flowfield surrounding a complete aircraft is partitioned into blocks, 3-dimensional volumes treating the fuselage, wings, and control surfaces. Parallelization of the computation occurs thru a domain decomposition strategy allocating one or more blocks to each processor. Each processor runs a copy of the flow solver and the various processors communicate with each other generally through nearest neighbor communication. Grid sizes and solution variables depend upon the specific problem, but a typical grid size might be a cube with 50 grid points on a side with 25 variables per grid point. The complicated geometry of the actual aircraft requires many grids and the need to adequately resolve the boundary layer demands that CFD codes operate on grids of this size.

NAS imposed no performance requirement on codes which executed on the SP2 and many of the SP2-executing codes were written on or for other machines with the multiprocessor versions made portable by employing PVM and/or MPI for interprocessor communication.

5. Workload Measurements

A single processor matrix multiply, fitting entirely in the 256 kB cache and fully blocked with the central loop unrolled, performs at approximately 240 Mflops on the 67 Mhz POWER2. This rate can be taken as an achievable single processor workload peak. The workload data also reflect the performance of multiprocessor message passing codes. The maximum multinode SP2 rate reported (but not measured by RS2HPM) for such a code is 29 Gflops. This code simulated electromagnetic scattering and relied heavily upon matrix (BLAS3) operations [Farhat,1996].

The NAS SP2 workload is highly variable in performance due to the different numbers of users and algorithms processed by the machine. During the period measured, there was no strong production component to the workload. Moreover, the distributed nature of the machine made it difficult to load all nodes with user jobs. The decision to give users dedicated access to the nodes also allowed the potential for additional system idle arising from message-passing and disk transfer related I/O delays.

Figure 1 shows the performance of the workload during the period from July of 1996 through March of 1997. The Figure shows the daily performance, the moving average of the daily performance and the moving average of the system utilization.The machine average utilization, defined as the fraction of elapsed time the SP2 nodes were servicing PBS jobs, was 64% during this period and the maximum daily utilization achieved during this period was 95%. The average daily system performance is about 1.3 Gflops on 144 processors. The average rate represents about 9 Mflops per processor or 3% of peak. A 24-hour rate of 3.4 Gflops was sustained in November 1996, and the maximum 15-minute rate measured during the nine-month period was 5.7 Gflops. The fluctuations shown in Figure 1 result more from load demand than code variability. Although the NAS administrators configured the SP2 for code development, the Figure shows no obvious trend toward increased performance as time passes.

To filter the effects of those days with high idle, we restrict our attention to days with performance exceeding 2.0 Gflops. For the 30 (of 270) days whose performance exceeded 2.0 Gflops, Table 2 reports the average and standard deviation for measured Mflops, Mips, and Mops rates along with representative rates for a single day. These rates represent single node values and system rates may be obtained by multiplying by 144.

This smaller SP2 sample displays a average performance level of 2.5 Gflops and a system utilization of 76% for the machine. This performance rate corresponds to about 1 FLOP every 4 cycles and to support this level of results, 45.7 Mips (memory instructions and branches) are required-about 1.5 instructions every cycle.

Table 2: Measured Major Rates for NAS Workload

Rates	Day 45.0	Avg Rate	Std
Mips	37.6	45.7	10.5
Mops	38.2	48.3	10.2
Mflops	17.0	17.4	3.8

Table 3 provides the breakdown of Mflops into floating-point adds, floating-point divides, floating-point adds, and floating-point multiply-adds. RS2HPM distinguishes between the floating-pointing operations executed by the compound fma instruction and those executed by a single instruction. The fma multiply appears in the fma operation count and the fma add appears in the add operation count. The fma instruction produces about 54% of the floating-point operations in the workload.

Table 3: Measured Major Rates for NAS Workload

Rates	Day 45.0	Avg	Std
		OPS	
Mflops-All	17.0	17.4	2.3
Mflops-add	10.2	9.5	1.5
Mflops-div	0.0	0.0	0.0
Mflops-mult	3.6	3.2	0.5
Mflops-fma	3.2	4.7	0.8
		INST	
Mips-Floating Point (Total)	16.4	14.8	2.0
Mips-Floating Point (Unit 0)	10.3	9.4	1.2
Mips-Floating Point (Unit 1)	6.1	5.4	0.8
Mips-Fixed Point Unit (Total)	18.8	27.6	5.8
Mips-Fixed Point (Unit 1	11.3	16.5	3.3
Mips-Fixed Point (Unit 0)	7.5	11.1	2.6
Mips-Inst Cache Unit	2.4	3.3	0.6
		CACHE	
Data Cache Misses-Million/S	0.30	0.30	0.06
TLB-Million/S	0.06	0.04	0.01
Instruction Cache Misses-Million/S	0.006	0.014	0.010
		I/O	
DMA reads-MTransfer/S	0.018	0.024	0.015
DMA writes-MTransfer/S	0.012	0.017	0.010

The POWER2 features dual generic FPUs and the HPM measurements show a distinct asymmetry between their floating-point rates. The instruction cache units dispatches floating-point instructions into a common queue which feeds the two floating-point units. Floating-point instructions are sent to FPU0 until the ICU encounters a dependency or attempts to perform a multicycle operation and then floating-point instructions are sent to FPU1. Multicycle operations include the 10-cycle divide and 15-cycle square root operations. Although a common instruction queue feeds both units, the POWER2 provides a backup register to provide buffering to allow one unit to continue while the second unit is processing such operations. The average ratio of instructions performed by FPU0 to those performed by FPU1 is 1.7 and while higher performance workloads should display ratios closer to 1, RS2HPM measurements on the NAS workload have yet to show such ratios. Add and multiply operations dominate typical CFD workloads, and it is unlikely that multicycle operations in one FPU are allowing other pipelines to drain. More likely is that the dependencies among the various instructions limit the amount of instruction-level parallelism available for exploitation.

Asymmetries also occur in the FXU measurements, but the differing design of the FXUs is responsible. FXU0 has additional responsibility in handling cache misses whereas FXU1 has the sole responsibility for performing the divide and multiply operations required for addressing operations. For simple test problems, the total number of instructions processed by the FXUs closely approximates the floating-point memory-to-register load / store operations. The average number of floating-point operations divided by the average number of floating-point memory instructions provides a good measure of the effectiveness of the code and compiler in register reuse. The average ratio for the small workload sample is 0.53; for comparison, the high performance matrix multiply displays a value of 3.0 for this ratio.The measurements indicate that workload codes in general do not yet make good use of the POWER2 registers.

The average instruction issue rate for the workload is 3.3 million instructions per second; this rate represents the rate at which the ICU fetches instructions from the instruction cache, dispatches instructions to the FPUs and FXUs, and instructions executed by the ICUs. In simple test problems, the branches at the end of DO-loops seem to dominate the number of instructions executed by the ICU. This interpretation indicates that about 11% of the instructions in the workload are branches.

Table 3 includes cache and TLB miss rates (per second) for the workload. We can use these to estimate cache miss ratios by dividing by the memory instruction issue rate. We approximate the memory instruction issue rate by the sum of FXU0 and FXU1. For well-written RISC codes, measurements indicate that this sum does give a good estimate of memory instructions, but generally this sum will include more than memory instructions. Using this sum gives a lower bound to the cache-miss ratio as 1.0% and a TLB-miss ratio as 0.1%. We can put these numbers into perspective by considering the case of sequentially accessing a single large array, with no cache reuse. The NAS SP2 processor had a cache line size of 256 bytes and a page size of 4096 bytes. For real*8 data, we would experience a cache-miss every 32 elements and a TLB miss rate every 512 elements. The NAS rates are comparable to such an access pattern.

Table 4: Hierarchical Memory Performance

Rate	NAS Workload	Sequential Access	NPB BT on 49 CPUs
Cache Miss Ratio	1%	3%	1.2%
TLB Miss Ratio	0.1%	0.2%	0.06%
Mflops/CPU	17		44

Table 4 shows these values along with the ratios reported by RS2HPM for the NAS Parallel Benchmark BT (Saphir, et al, 1996). The low value for the BT TLB miss ratio reflects the efficient memory access pattern obtained by rearranging the main loop nests to access memory in a way that promoted cache reuse.

We might expect high TLB miss rates from programs accessing data with large memory strides.

A program referencing data not in cache will take a cache miss and execution may halt for 8 cycles while the reference is satisfied by bringing in the appropriate data into the cache line. If the data is not on a page residing in memory, a TLB miss occurs and the processor may experience a delay of 36 to 54 cycles until the reference is satisfied. We can quantify this delay by expressing it as delay per memory instruction, approximating memory references by the instructions reported in the FXU0 and FXU1 counters. The exact number of memory references is unknown since the Table 1 counter selection allows

RS2HPM to count a quad load or quad store as a single instruction. The delay per memory reference is about 0.12 cycle per memory reference.

The table reports the instruction cache miss rate as 0.014 million per second which means about 0.4% of the instruction fetches experience a cache miss. This rate is low because most of the branches in typical floating-point loops return control back to the top of the loop and re-execute the same instructions.

The DMA measurements represent transfers per second and a single transfer can represent either 4 or 8 words. Most of the DMA traffic represents message-passing I/O and the measured rate of 0.042e6 reads and writes corresponds to about 1.3 Mbytes/second which is about 4% of the network node-to-node bandwidth. During the period monitored, the maximum network node-to-node rate sustained during a 15-minute period was 5.4 Mbytes/second, corresponding to 16% of peak. The Table 1 counter selection does not allow a distinction between message-passing I/O and disk I/O, but measurements indicate that disk traffic appears in the system report of the DMA read/write and the average value for disk I/O traffic is 3.2 Mbytes/second.

There were no obvious trends in the RS2HPM workload data, as might be expected in a machine capable of performing calculations at two different rates such as a vector machine [Williams,1988]. For example, workloads executing a greater fraction of floating-point operations in the fma unit should display a higher performance rate, but NAS workload measurements have yet to display such a trend. The lack of obvious trends such as reductions in performance rates with increasing cache and/or TLB miss rates is difficult to analyze since the NAS 22-counter selection excluded performance reducing factors such as message-passing delays and I/O wait times.

6. Batch Job Measurements

Modifications to the SP2 batch system, PBS, allow the RS2HPM allows to record the performance of individual batch jobs. Users and system personnel may examine and analyze the hardware counts reported for these jobs. To reduce the impact of the interactive sessions, this discussion examines only jobs exceeding 600 seconds of wall clock time. This restriction also has the property of removing many of the non-user benchmarking codes. The time-weighted average for the jobs in this database was 19 Mflops per node.

Figure 2, the distribution of batch job wall clock time according to the number of nodes requested, shows that moderately parallel 16, 32, and 8-node jobs consumed most of the wall clock time. The figure shows essentially no wall clock time consumed by jobs requesting more than 64 nodes.System administrators could not checkpoint MPI/PVM jobs and had to rely upon draining the queues to allow jobs requesting more than 64-nodes to execute. Even when such jobs executed, they did not consume significant wallclock time.

Figure 3 shows the performance per node of jobs again as a function of the number of nodes requested. While there is a sharp decrease in performance beyond 64 nodes, the per node batch job rate is sustained in many cases up to 64 nodes. The peak rate of approximately 40 Mflops per node on 28 nodes involved a Navier-Stokes solver with each of the 28 nodes computing on a 96x96x32 grid. This application employed a domain decomposition to obtain a geometry-based parallelism and used asynchronous message-passing (Cui and Street,1997).

Since the intent of the machine was to promote algorithm development and since users would presumably improve performance over time, it is reasonable to examine the history of jobs grouped by node. Figure 4 shows the performance of batch jobs requesting 16-nodes, the most popular selection, as a function of batch job id. The average value is 320 Mflops with a variance of 200 Mflops. While the performance spread is quite high, the moving average indicates no trend toward improvement as time passes. Similar trends occur for other processor counts.

The per node performance of the SP2 batch jobs degrades seriously as the number of nodes increase beyond 64. A few of the user jobs requesting such a large number of nodes were not floating-point intensive and others were using synchronous communication. HPM output for the remaining jobs using more than 64-nodes indicated that the instructions issued by the FXU and ICU while the processor was in system mode exceeded those issued while the processor was in user mode. Evidently these processes were paging data, and discussions with the users confirmed this suspicion. Re-examination of the workload data, as shown in Figure 5,confirmed that high system intervention occurred on days displaying below average global performance.

Enforcement of a no-paging data restriction on the compute nodes would require considerable rewriting of the current batch system scheduler. Many of the codes employ automatic arrays whose memory requirements appear only at runtime. There is currently no diagnostic on the SP2 to inform a user of such data paging, short of logging onto the nodes at the time of execution. Users may not be willing to reallocate their scarce real-time resources to repackage their codes to avoid paging.

7. Conclusions

The RS2HPM system reports that the NAS SP2 delivers about 1.3 Gflops daily on a floating-point intensive CFD workload for an overall system efficiency of about 3%. Measurements showed no tendency for this performance to increase over time despite the fact that the NAS SP2 administrators provided an environment for algorithm development. Workload measurements did show a high rate of system overhead for days during which performance was poor and batch measurements showed a similar rate of overhead for poorly performing jobs. The source of this problem appears to be a large amount of data paging induced by node memory oversubscription.That such paging significantly detracted from workload performance was a surprising finding.

The individual batch job measurements indicate that many of the users have not rewritten their codes to take advantage of POWER2 performance features. The ratio of flops to memory references was 1.0, indicating that many of the codes were not making good reuse of the registers. About 50% of the workload floating-point operations resulted from the fma instruction, but the better-performing individual codes perform at least 80% of their operations from fma instructions. Measurements also show relatively high TLB miss rates.

The IBM POWER2 monitor has been quite effective in diagnosing some of the reasons for this performance level. The monitor also allows a confirmation or denial of anecdotal reports of system performance. This tool has identified suboptimal usage of the cache as manifested by high TLB miss rates and a high degree of paging. The ability to monitor the amount of intervention by the operating system was a very useful feature. Other sites wishing to monitor their SP or SP2 systems might consider selecting counter options which could also report I/O wait time in addition to CPU performance.

References

A. Cui and R. Street, “Parallel Computing of Upwelling in a Rotating Stratified Flow”, Proc. 50th Annual Mtg of the Fluid Dynamics Division of the American Physical Society, 1997.

C. Farhat, “Large, Out-of-Core Calculation Runs on the IBM SP2,” NAS News, **2**,11,1995.

Jussi Maki,”POWER2 Hardware Performance Tools”, URL http://www.csc.fi/~jmaki/rs2hpm_paper.

S. Saiyed, et al.”POWER2 CPU-Intensive Workload Performance,” URL http://www.rs6000.ibm.com/resource/technology/SPEC.html

W. Saphir, Alex Woo, and Maurice Yarrow, ”The NAS Parallel Benchmarks 2.1 Results”, NAS Report NAS-96-010, August 1996.

W. Saphir, ”PHPM: Parallel Hardware Performance Monitoring for the IBM-SP2,” NAS Internal Memorandum, October 1996.

C. B. Stunkel, et al,”The SP2 High-Performance Switch”, IBM Systems Journal, **34**, No. 2, 1995.

E.H. Welbon, et al.,”The POWER2 Performance Monitor,” IBM J. Res. Develop.,**38**, No. 5, 545-554, September 1994.

White, S. W. and Dhawan,S.,“POWER2:Next generation of the RISC System/6000 family,” IBM J. Res. Develop.,**38**, No. 5, 493-502, September 1994.

E.Williams and R.Koskela. 1988. Measurement of a scientific workload using the Cray X-MP performance monitor. In Proc. 21st Cray User Group Mtg: 411-422.

Figure 1

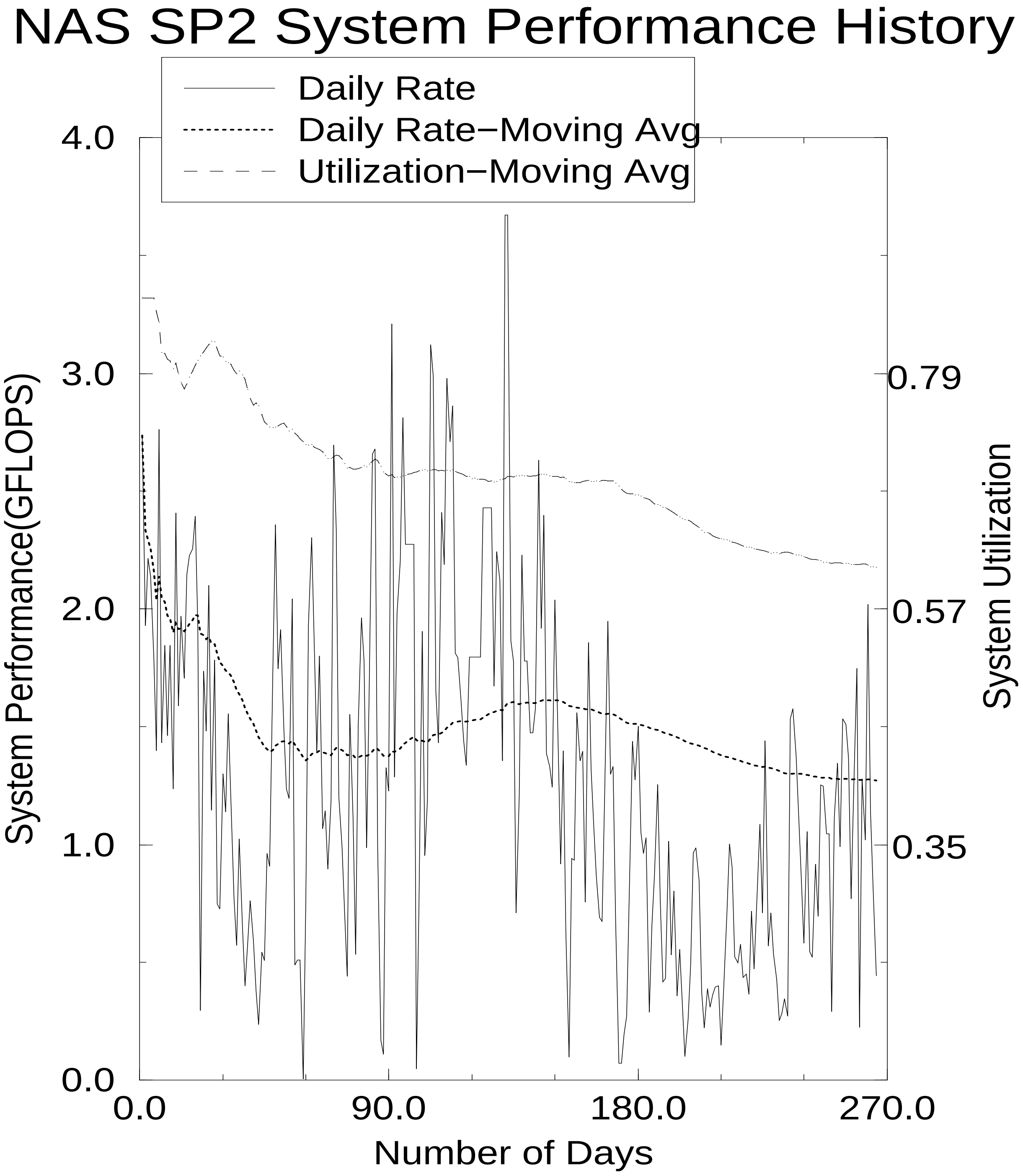

NAS SP2 System Performance History
Daily Rate
Daily Rate−Moving Avg
Utilization−Moving Avg
4.0
3.0
2.0
1.0
0.0
0.79
0.57
0.35
0.0
90.0
180.0
270.0
Number of Days
System Performance(GFLOPS)
System Utilization

Figure 2

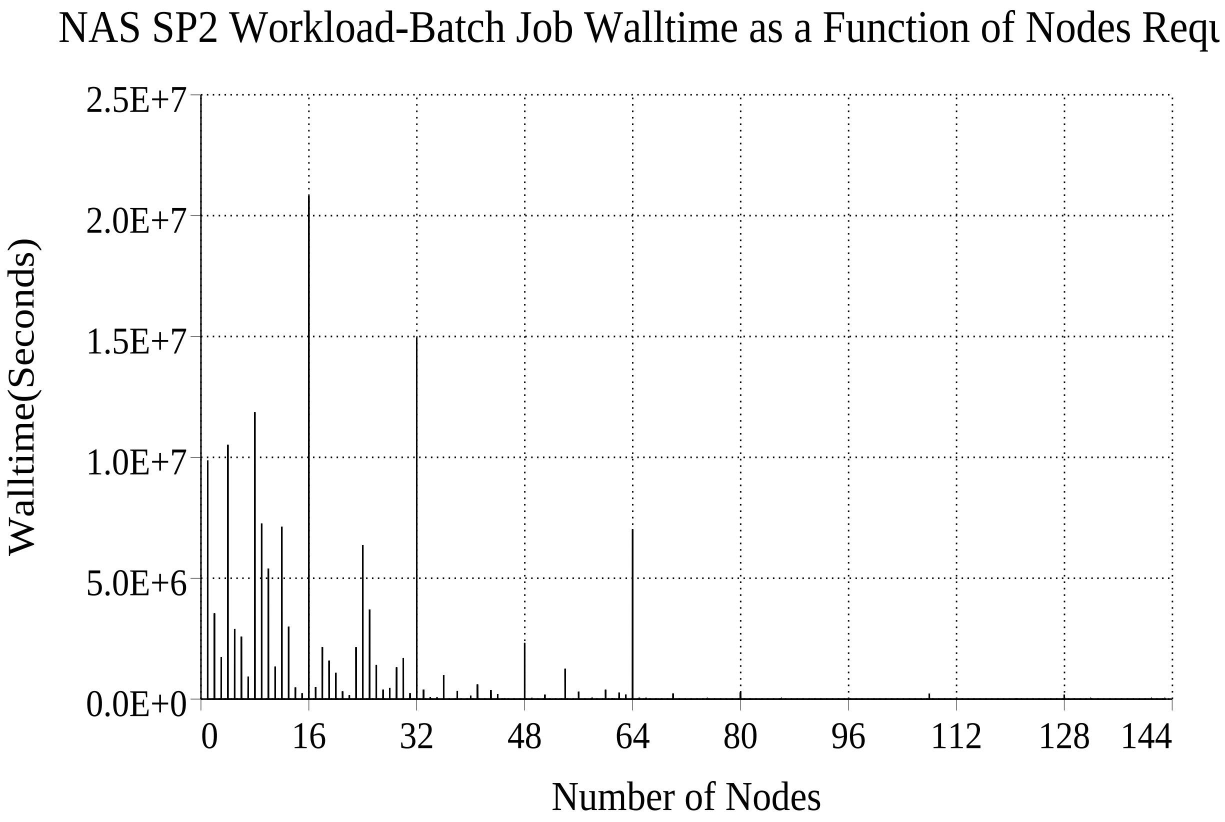

NAS SP2 Workload-Batch Job Walltime as a Function of Nodes Requ
Walltime(Seconds)
2.5E+7
2.0E+7
1.5E+7
1.0E+7
5.0E+6
0.0E+0
0
16
32
48
64
80
96
112
128
144
Number of Nodes

Figure 3

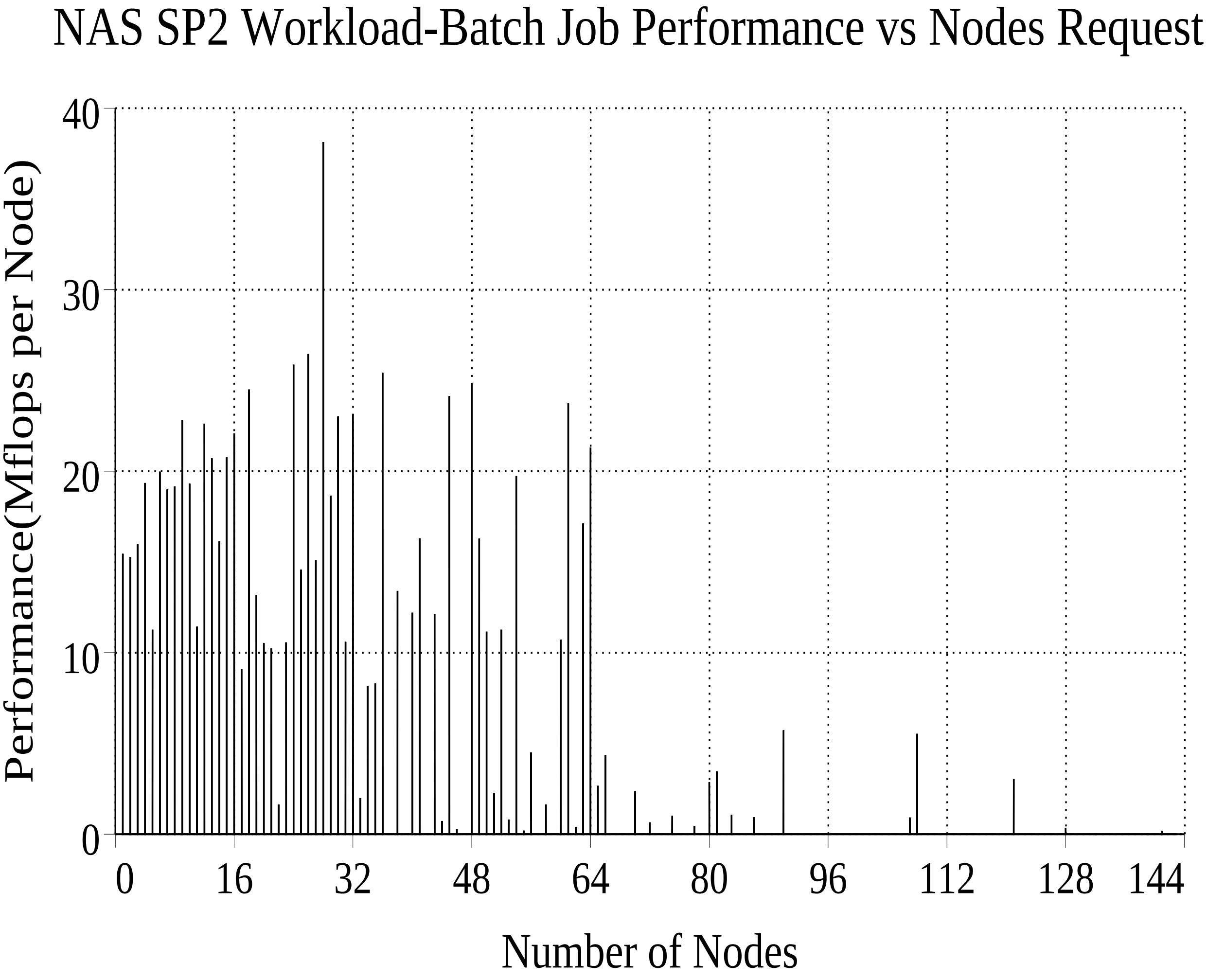

NAS SP2 Workload-Batch Job Performance vs Nodes Request
Performance(Mflops per Node)
0
10
20
30
40
0
16
32
48
64
80
96
112
128
144
Number of Nodes

Figure 4

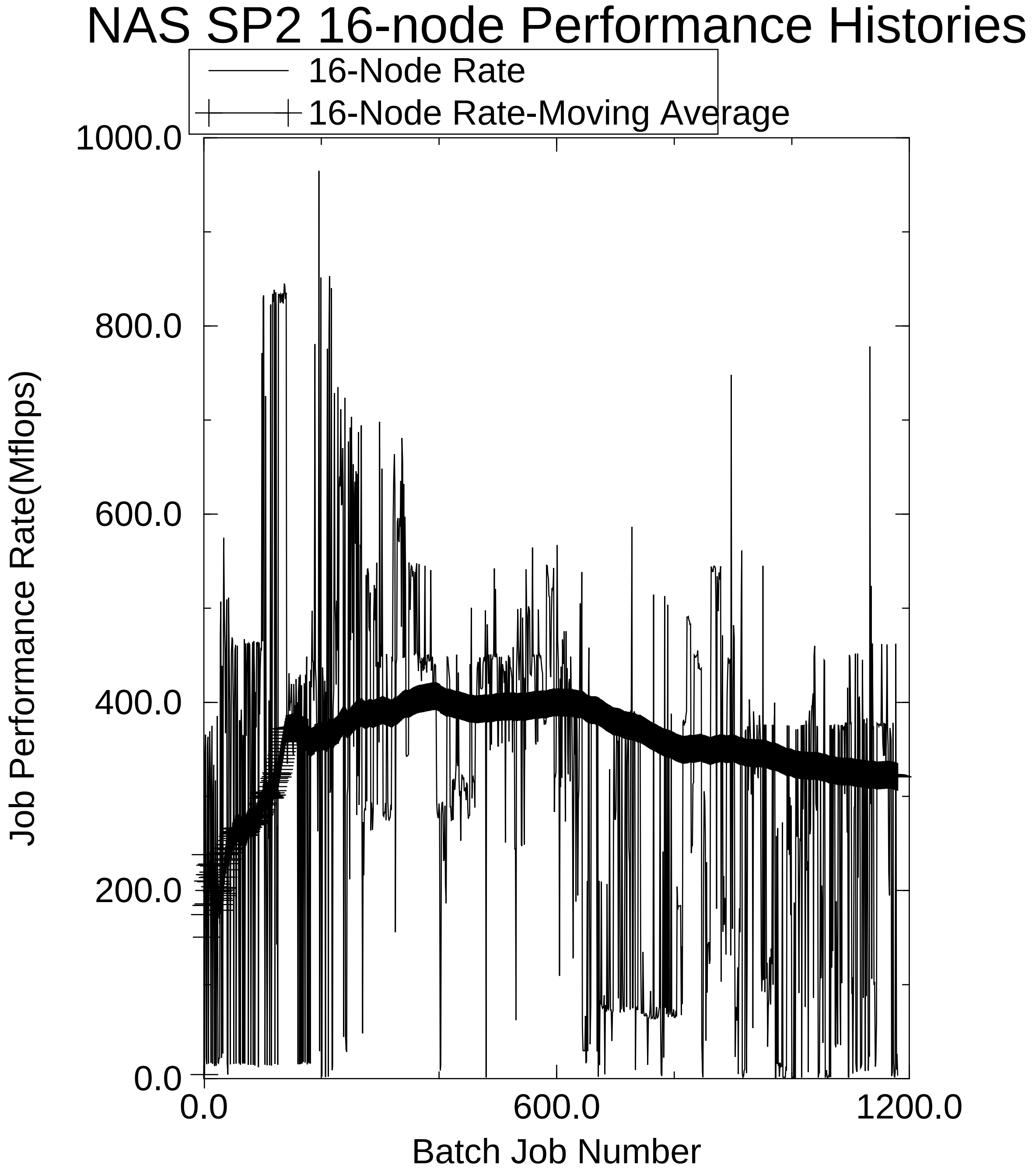
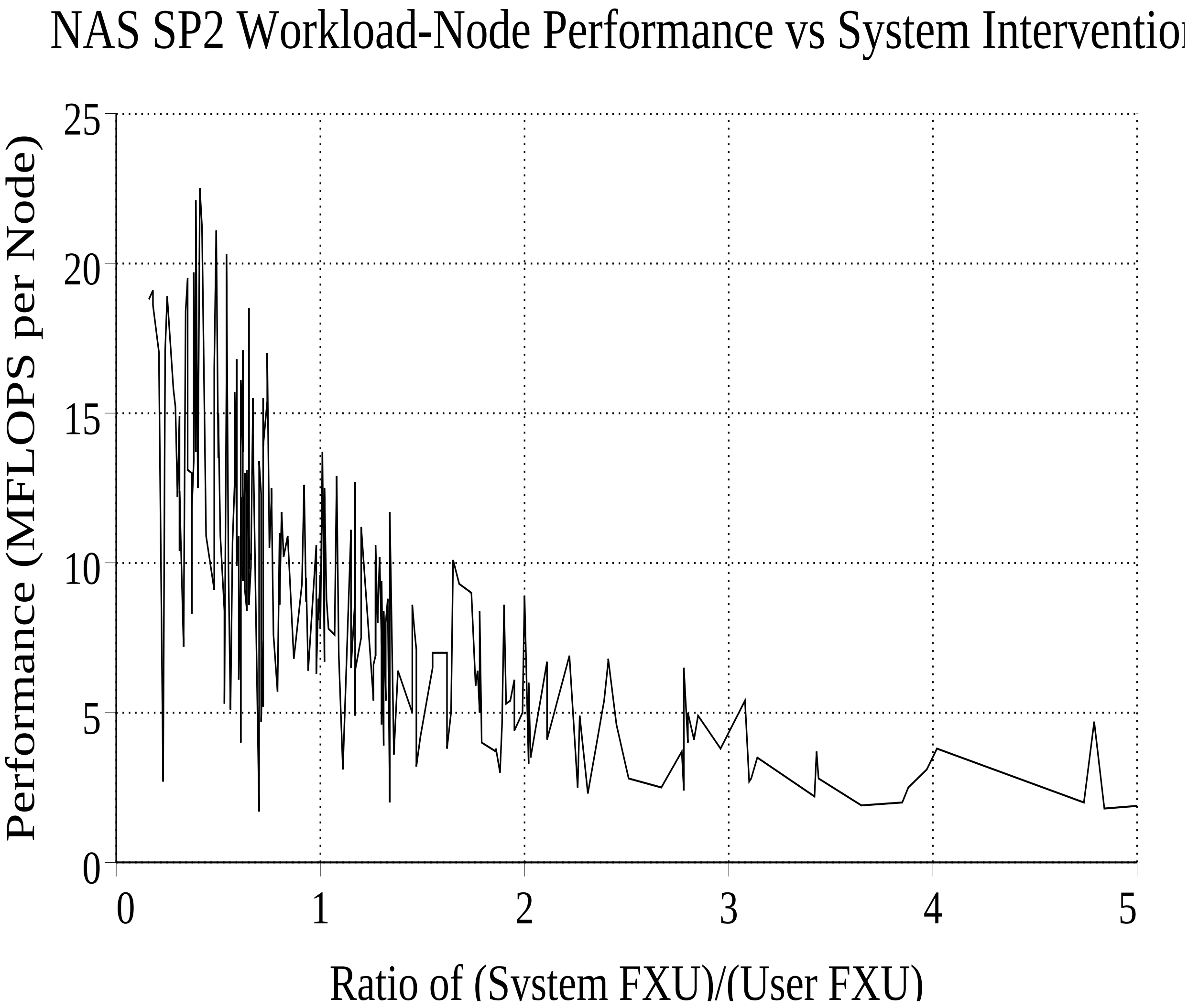

NAS SP2 16-node Performance Histories
16-Node Rate
16-Node Rate-Moving Average
1000.0
800.0
600.0
400.0
200.0
0.0
Job Performance Rate(Mflops)
0.0
600.0
1200.0
Batch Job Number

Figure 5


NAS SP2 Workload-Node Performance vs System Interventio
Performance (MFLOPS per Node)
25
20
15
10
5
0
0
1
2
3
4
5
Ratio of (System FXU)/(User FXU)

Measurement of a Scientific Workload using the IBM Hardware Performance Monitor

Robert J. Bergeron
bergeron@nas.nasa.gov

NAS Systems Division
NASA Ames Research Center
Mail Stop 258-5
Moffett Field, CA 94035-1000

Abstract

This paper presents data on the performance of the NAS SP2 IBM system using RS6000 hardware monitors and a performance measurement tool. The data collected showed that the SP2 averages about 1.3 Gflops, about 3% of peak. The report provides the relative usage for the various hardware units over the entire workload measured over a 9-month period. The workload displays moderate parallelism, with the most popular choice of nodes as 16. Although the monitor data provide a good snapshot of workload performance, causal correlations regarding key performance indicators appear difficult to draw from the current data due to the absence of I/O delay measurements.

1. Introduction

For many years, supercomputers have employed hardware monitors designed and built into their custom processors to display individual user code characteristics and total system performance.Recently, RISC processors have begun to supply hardware monitors which are software accessible to users.
In the 1995-1996 period, NAS used a combination of RS6000 hardware monitors and a performance measurement tool to monitor the performance of its floating-point, memory intensive workload on a cluster of IBM RISC POWER2 processors. This monitor and the software which provides an interface to the monitor counters permit a detailed description of the CPU instructions executed, counts and delays associated with cache and TLB misses, and utilization of the various execution elements. This paper will describe the computer system used to execute the workload, the monitor used to report performance characteristics, and some general characteristics of this workload. Then the paper will describe the workload measurements, the characteristics of the batch jobs, and the degree of parallelism in this workload. Finally, the paper will provide some remarks regarding the performance monitoring of parallel systems.

2.Description of the System

The NAS SP2 is a distributed memory 144-node RS6000/590 cluster connected by a high-performance proprietary IBM network. The RS6000 nodes comprising the cluster consist of semi-custom chips or units described briefly as follows [White and Dhawan,1994]:

o-The Instruction Decode Unit(ICU)
This unit prefetches instructions from the instruction cache. The ICU executes branch and condition register instructions and dispatches instructions to the Fixed Point Units and the Floating Point Units. The ICU can prefetch 8 instructions/cycle from the instruction cache and can dispatch 4 instructions/

cycle to the Fixed Point Units and the Floating Point Unit

o-The Fixed Point Unit(FXU)
This dual unit (FXU0 and FXU1) processes all storage references, integer arithmetic, and logical operations. The FXU can execute 2 instructions (including 2 storage references) per cycle. The unit generates addresses in its general purpose registers(GPRs). The FXU also performs data cache directory searches.

o-The Floating Point Unit(FPU)
This dual unit (FPU0 and FPU1) contains Floating Point Registers(FPRs) and 2 64-bit execution units. When executing a compound floating point multiply add instruction (fma), the FPU can produce 4 floating point operations(FLOPS) per cycle. The FPU also contains hardware to overlap floating point stores with its arithmetic operations.

o-The Data Cache Unit(DCU)
The DCU supports two one-word data busses to the FXU, two quad-word busses to the FPU, a four-word instruction bus to the ICU and a two-word system I/O (SIO) bus to the I/O subsystem for Direct Memory Access(DMA) support. This cache is a four-way set-associative dual ported cache consisting of four identical chips. RISC processors employ a memory hierarchy to transport data from the off-chip memory to the cache and then to the registers.

o-The Storage Control Unit(SCU)
This unit controls communications between the CPU, the memory and the SIO bus.

o-The I/O Unit(SCU)
This unit controls the I/O by implementing a 64-bit streaming data protocol on the Power2 data line, the Micro Channel.

Each node has at least 128 Mbytes of main memory and 2 Gbytes of disk space. The NAS SP2 processors provide a 4-way set associative data cache of 256 kB, arranged in 1024 lines of 256 bytes each. The IBM RISC 6000 implements virtual memory with a page size of 4096 bytes and supports 512 entries in the TLB. Each of the 144 nodes executes a copy of the IBM version of UNIX. For the period monitored, this version was AIX 4.1.3. The SP2 processor operates at a clock rate of 66.7 Mhz, and displays a peak performance of 267 Mflops.
The nodes were interconnected by the High Performance Switch (Stunkel et al, 1995) through communication adapters attached to the node input/output bus. This network displayed a latency of approximately 45 microseconds and a bandwidth of 34 Mbyte/second. The available communication bandwidth over this switch scales linearly with the number of processors. Extensive testing at NAS indicated this switch allowed a variety of parallel applications to scale well and the system displayed little performance degradation when tested under a full load of message-passing jobs.
The NAS SP2 provided an NFS-mounted external filesystem accessible by all nodes with 3 home filesystems of 8 GB each. Data transfers from the SP2 nodes to the home filesystems also occurred over the switch.
NAS employed its Portable Batch System(PBS) for job management. Key features of PBS included support for parallel job scheduling and direct enforcement of resource allocation policies. PBS also provides interactive login to the SP2 nodes, which allows users to more easily debug message-passing programs.

3. Description of the Monitor

The SP2 POWER2 Performance Monitor consist of 22 32-bit counters located on the SCU chip which can report CPU and storage-related events. The POWER2 counters provide a set of 5 counters and 16 reportable events each for the FPU, the FXU, the ICU, and the SCU. The selected 22 events are a subset of the 320(some overlapping) signals which can be selected and reported by software [Welbon,1994].

-NAS Counter Selection

The hardware monitor allows many possible combinations of events, but each combination must be implemented and verified in the monitoring software.The NAS counter selections, shown in Table 1, were chosen to give a broad overview of workload CPU performance.

Table 1: NAS SP2 RS2HPM Counters

Counter	Label	Description
user.fxu0	FXU[0]	number of instructions executed by Execution unit 0
user.fxu1	FXU[1]	number of instructions executed by Execution unit 1
user.dcache_mis	FXU[2]	FPU and FXU requests for data not in the D-cache
user.tlb_mis	FXU[3]	FPU and FXU requests for data not in the D-cache
user.cycles	FXU[4]	user cycles
user.fpu0	FPU0[0]	arithmetic instructions executed by Math 0
fpop.fp_add	FPU0[1]	floating point adds executed by Math 0
fpop.fp_mul	FPU0[2]	floating point multiplies executed by Math 0
fpop.fp_div	FPU0[3]	floating point divides executed by Math 0
fpop.fp_muladd	FPU0[4]	floating point multiply-adds executed by Math 0
user.fpu1	FPU1[0]	arithmetic instructions executed by Math 1
fpop.fp_add	FPU1[1]	floating point adds executed by Math 1
fpop.fp_mul	FPU1[2]	floating point multiplies executed by Math 1
fpop.fp_div	FPU1[3]	floating point divides executed by Math 1
fpop.fp_muladd	FPU1[4]	floating point multiply-adds executed by Math 1
user.icu0	ICU[0]	number of type I instructions executed
user.icu1	ICU[1]	number of type II instructions executed
user.icache_reload	SCU[0]	data transfers from memory to the I-cache
user.dcache_reload	SCU[1]	data transfers from memory to the D-cache
user.dcache_store	SCU[2]	number of transfers of D-cache data to memory
		occurs when the D-cache destination for incoming data currently contains data which has been modified
user.dma_read	SCU[3]	data transfers from memory to an I/O device
user.dma_write	SCU[4]	data transfers to memory from an I/O device

The monitor reports floating-point adds, multiplies, and fma operations. The fma operation counts as an add and a multiply for the purpose of flop counting. An implementation error in the hardware monitor prevented the proper reporting of the division operations, which typically constitute about 3% of the total floating operations for the NAS workloads. The monitor also reports the number of instructions issued by the floating-point units. The instructions issued by the fixed point units are predominantly memory loads and stores for the CFD codes in the NAS workload.The monitor also reports instruction and data cache reloads and misses.The Direct Memory Access (DMA) counters report the level of I/O activity for data moved between memory and the I/O devices; these counters also report the amount of I/O associated with message-passing.

IBM did not distribute software to access the RS6000 hardware counters, but in 1995 Jussi Maki of the Center for Scientific Computing in Finland created a set of tools for monitoring the POWER2 hardware counters [Maki,1995]. These tools allowed the reporting of events occurring in both user and system mode thru a multipass sampling mode. After NAS had obtained IBM's approval, Bill Saphir(NERSC) installed these tools on the SP2. These tools, collectively termed RS2HPM herein, consist of library, data collection daemon, kernel extension and other utilities. Bill Saphir also introduced some valuable extensions of these tools to allow monitoring of individual job performance, as well as global system performance [Saphir,1996].

-System-wide data collection

The RS2HPM daemon, executing on all nodes of the SP2, allows automatic sampling and data access over the network via TCP. At 15-minutes intervals, the cron daemon runs a script to collect data from all the SP2 nodes which are available for user jobs and stores this data for later analysis. This daemon collects performance data from the nodes whether or not user processes are executing.

-Batch job data collection

The PBS batch system runs a prologue script before each job and an epilogue script after each job. These scripts know which SP2 nodes the batch job is using and obtain counter values at the beginning and end of each job for these nodes. These values are written to a file for later processing and viewing by both users and system personnel. For individual programs to be reported, users must place commands into their batch scripts or preface interactive sessions with the appropriate RS2HPM commands.

4. Workload Description

The NAS workload consists primarily of codes solving computational fluid dynamics problems involving aerodynamics, hypersonics, propulsion, and turbulence. Many of the aerodynamics workload codes perform parallel multidisciplinary optimization which involves systematically modifying an aircraft configuration to maximize or minimize a chosen aerodynamic figure of merit. This approach involves coupling a CFD solver to a numerical optimization procedure and should display a high degree of parallelism since computations on the various configurations are completely independent.

Other aerodynamics codes, constituting the majority of the NAS SP2 workload, involve multiple grids treating a single aircraft. There are a variety of numerical methods for treating such problems, but most would involve the following steps. The flowfield surrounding a complete aircraft is partitioned into blocks, 3-dimensional volumes treating the fuselage, wings, and control surfaces. Parallelization of the computation occurs thru a domain decomposition strategy allocating one or more blocks to each processor. Each processor runs a copy of the flow solver and the various processors communicate with each other generally through nearest neighbor communication. Grid sizes and solution variables depend upon the specific problem, but a typical grid size might be a cube with 50 grid points on a side with 25 variables per grid point. The complicated geometry of the actual aircraft requires many grids and the need to adequately resolve the boundary layer demands that CFD codes operate on grids of this size.

NAS imposed no performance requirement on codes which executed on the SP2 and many of the SP2-executing codes were written on or for other machines with the multiprocessor versions made portable by employing PVM and/or MPI for interprocessor communication.

5. Workload Measurements

A single processor matrix multiply, fitting entirely in the 256 kB cache and fully blocked with the central loop unrolled, performs at approximately 240 Mflops on the 67 Mhz POWER2. This rate can be taken as an achievable single processor workload peak. The workload data also reflect the performance of multiprocessor message passing codes. The maximum multinode SP2 rate reported (but not measured by RS2HPM) for such a code is 29 Gflops. This code simulated electromagnetic scattering and relied heavily upon matrix (BLAS3) operations [Farhat,1996].

The NAS SP2 workload is highly variable in performance due to the different numbers of users and algorithms processed by the machine. During the period measured, there was no strong production component to the workload. Moreover, the distributed nature of the machine made it difficult to load all nodes with user jobs. The decision to give users dedicated access to the nodes also allowed the potential for additional system idle arising from message-passing and disk transfer related I/O delays.

Figure 1 shows the performance of the workload during the period from July of 1996 through March of 1997. The Figure shows the daily performance, the moving average of the daily performance and the moving average of the system utilization.The machine average utilization, defined as the fraction of elapsed time the SP2 nodes were servicing PBS jobs, was 64% during this period and the maximum daily utilization achieved during this period was 95%. The average daily system performance is about 1.3 Gflops on 144 processors. The average rate represents about 9 Mflops per processor or 3% of peak. A 24-hour rate of 3.4 Gflops was sustained in November 1996, and the maximum 15-minute rate measured during the nine-month period was 5.7 Gflops. The fluctuations shown in Figure 1 result more from load demand than code variability. Although the NAS administrators configured the SP2 for code development, the Figure shows no obvious trend toward increased performance as time passes.

To filter the effects of those days with high idle, we restrict our attention to days with performance exceeding 2.0 Gflops. For the 30 (of 270) days whose performance exceeded 2.0 Gflops, Table 2 reports the average and standard deviation for measured Mflops, Mips, and Mops rates along with representative rates for a single day. These rates represent single node values and system rates may be obtained by multiplying by 144.

This smaller SP2 sample displays a average performance level of 2.5 Gflops and a system utilization of 76% for the machine. This performance rate corresponds to about 1 FLOP every 4 cycles and to support this level of results, 45.7 Mips (memory instructions and branches) are required-about 1.5 instructions every cycle.

Table 2: Measured Major Rates for NAS Workload

Rates	Day 45.0	Avg Rate	Std
Mips	37.6	45.7	10.5
Mops	38.2	48.3	10.2
Mflops	17.0	17.4	3.8

Table 3 provides the breakdown of Mflops into floating-point adds, floating-point divides, floating-point adds, and floating-point multiply-adds. RS2HPM distinguishes between the floating-pointing operations executed by the compound fma instruction and those executed by a single instruction. The fma multiply appears in the fma operation count and the fma add appears in the add operation count. The fma instruction produces about 54% of the floating-point operations in the workload.

Table 3: Measured Major Rates for NAS Workload

Rates	Day 45.0	Avg	Std
		OPS	
Mflops-All	17.0	17.4	2.3
Mflops-add	10.2	9.5	1.5
Mflops-div	0.0	0.0	0.0
Mflops-mult	3.6	3.2	0.5
Mflops-fma	3.2	4.7	0.8
		INST	
Mips-Floating Point (Total)	16.4	14.8	2.0
Mips-Floating Point (Unit 0)	10.3	9.4	1.2
Mips-Floating Point (Unit 1)	6.1	5.4	0.8
Mips-Fixed Point Unit (Total)	18.8	27.6	5.8
Mips-Fixed Point (Unit 1	11.3	16.5	3.3
Mips-Fixed Point (Unit 0)	7.5	11.1	2.6
Mips-Inst Cache Unit	2.4	3.3	0.6
		CACHE	
Data Cache Misses-Million/S	0.30	0.30	0.06
TLB-Million/S	0.06	0.04	0.01
Instruction Cache Misses-Million/S	0.006	0.014	0.010
		I/O	
DMA reads-MTransfer/S	0.018	0.024	0.015
DMA writes-MTransfer/S	0.012	0.017	0.010

The POWER2 features dual generic FPUs and the HPM measurements show a distinct asymmetry between their floating-point rates. The instruction cache units dispatches floating-point instructions into a common queue which feeds the two floating-point units. Floating-point instructions are sent to FPU0 until the ICU encounters a dependency or attempts to perform a multicycle operation and then floating-point instructions are sent to FPU1. Multicycle operations include the 10-cycle divide and 15-cycle square root operations. Although a common instruction queue feeds both units, the POWER2 provides a backup register to provide buffering to allow one unit to continue while the second unit is processing such operations. The average ratio of instructions performed by FPU0 to those performed by FPU1 is 1.7 and while higher performance workloads should display ratios closer to 1, RS2HPM measurements on the NAS workload have yet to show such ratios. Add and multiply operations dominate typical CFD workloads, and it is unlikely that multicycle operations in one FPU are allowing other pipelines to drain. More likely is that the dependencies among the various instructions limit the amount of instruction-level parallelism available for exploitation.

Asymmetries also occur in the FXU measurements, but the differing design of the FXUs is responsible. FXU0 has additional responsibility in handling cache misses whereas FXU1 has the sole responsibility for performing the divide and multiply operations required for addressing operations. For simple test problems, the total number of instructions processed by the FXUs closely approximates the floating-point memory-to-register load / store operations. The average number of floating-point operations divided by the average number of floating-point memory instructions provides a good measure of the effectiveness of the code and compiler in register reuse. The average ratio for the small workload sample is 0.53; for comparison, the high performance matrix multiply displays a value of 3.0 for this ratio.The measurements indicate that workload codes in general do not yet make good use of the POWER2 registers.

The average instruction issue rate for the workload is 3.3 million instructions per second; this rate represents the rate at which the ICU fetches instructions from the instruction cache, dispatches instructions to the FPUs and FXUs, and instructions executed by the ICUs. In simple test problems, the branches at the end of DO-loops seem to dominate the number of instructions executed by the ICU. This interpretation indicates that about 11% of the instructions in the workload are branches.

Table 3 includes cache and TLB miss rates (per second) for the workload. We can use these to estimate cache miss ratios by dividing by the memory instruction issue rate. We approximate the memory instruction issue rate by the sum of FXU0 and FXU1. For well-written RISC codes, measurements indicate that this sum does give a good estimate of memory instructions, but generally this sum will include more than memory instructions. Using this sum gives a lower bound to the cache-miss ratio as 1.0% and a TLB-miss ratio as 0.1%. We can put these numbers into perspective by considering the case of sequentially accessing a single large array, with no cache reuse. The NAS SP2 processor had a cache line size of 256 bytes and a page size of 4096 bytes. For real*8 data, we would experience a cache-miss every 32 elements and a TLB miss rate every 512 elements. The NAS rates are comparable to such an access pattern.

Table 4: Hierarchical Memory Performance

Rate	NAS Workload	Sequential Access	NPB BT on 49 CPUs
Cache Miss Ratio	1%	3%	1.2%
TLB Miss Ratio	0.1%	0.2%	0.06%
Mflops/CPU	17		44

Table 4 shows these values along with the ratios reported by RS2HPM for the NAS Parallel Benchmark BT (Saphir, et al, 1996). The low value for the BT TLB miss ratio reflects the efficient memory access pattern obtained by rearranging the main loop nests to access memory in a way that promoted cache reuse.

We might expect high TLB miss rates from programs accessing data with large memory strides.

A program referencing data not in cache will take a cache miss and execution may halt for 8 cycles while the reference is satisfied by bringing in the appropriate data into the cache line. If the data is not on a page residing in memory, a TLB miss occurs and the processor may experience a delay of 36 to 54 cycles until the reference is satisfied. We can quantify this delay by expressing it as delay per memory instruction, approximating memory references by the instructions reported in the FXU0 and FXU1 counters. The exact number of memory references is unknown since the Table 1 counter selection allows

RS2HPM to count a quad load or quad store as a single instruction. The delay per memory reference is about 0.12 cycle per memory reference.

The table reports the instruction cache miss rate as 0.014 million per second which means about 0.4% of the instruction fetches experience a cache miss. This rate is low because most of the branches in typical floating-point loops return control back to the top of the loop and re-execute the same instructions.

The DMA measurements represent transfers per second and a single transfer can represent either 4 or 8 words. Most of the DMA traffic represents message-passing I/O and the measured rate of 0.042e6 reads and writes corresponds to about 1.3 Mbytes/second which is about 4% of the network node-to-node bandwidth. During the period monitored, the maximum network node-to-node rate sustained during a 15-minute period was 5.4 Mbytes/second, corresponding to 16% of peak. The Table 1 counter selection does not allow a distinction between message-passing I/O and disk I/O, but measurements indicate that disk traffic appears in the system report of the DMA read/write and the average value for disk I/O traffic is 3.2 Mbytes/second.

There were no obvious trends in the RS2HPM workload data, as might be expected in a machine capable of performing calculations at two different rates such as a vector machine [Williams,1988]. For example, workloads executing a greater fraction of floating-point operations in the fma unit should display a higher performance rate, but NAS workload measurements have yet to display such a trend. The lack of obvious trends such as reductions in performance rates with increasing cache and/or TLB miss rates is difficult to analyze since the NAS 22-counter selection excluded performance reducing factors such as message-passing delays and I/O wait times.

6. Batch Job Measurements

Modifications to the SP2 batch system, PBS, allow the RS2HPM allows to record the performance of individual batch jobs. Users and system personnel may examine and analyze the hardware counts reported for these jobs. To reduce the impact of the interactive sessions, this discussion examines only jobs exceeding 600 seconds of wall clock time. This restriction also has the property of removing many of the non-user benchmarking codes. The time-weighted average for the jobs in this database was 19 Mflops per node.

Figure 2, the distribution of batch job wall clock time according to the number of nodes requested, shows that moderately parallel 16, 32, and 8-node jobs consumed most of the wall clock time. The figure shows essentially no wall clock time consumed by jobs requesting more than 64 nodes.System administrators could not checkpoint MPI/PVM jobs and had to rely upon draining the queues to allow jobs requesting more than 64-nodes to execute. Even when such jobs executed, they did not consume significant wallclock time.

Figure 3 shows the performance per node of jobs again as a function of the number of nodes requested. While there is a sharp decrease in performance beyond 64 nodes, the per node batch job rate is sustained in many cases up to 64 nodes. The peak rate of approximately 40 Mflops per node on 28 nodes involved a Navier-Stokes solver with each of the 28 nodes computing on a 96x96x32 grid. This application employed a domain decomposition to obtain a geometry-based parallelism and used asynchronous message-passing (Cui and Street,1997).

Since the intent of the machine was to promote algorithm development and since users would presumably improve performance over time, it is reasonable to examine the history of jobs grouped by node. Figure 4 shows the performance of batch jobs requesting 16-nodes, the most popular selection, as a function of batch job id. The average value is 320 Mflops with a variance of 200 Mflops. While the performance spread is quite high, the moving average indicates no trend toward improvement as time passes. Similar trends occur for other processor counts.

The per node performance of the SP2 batch jobs degrades seriously as the number of nodes increase beyond 64. A few of the user jobs requesting such a large number of nodes were not floating-point intensive and others were using synchronous communication. HPM output for the remaining jobs using more than 64-nodes indicated that the instructions issued by the FXU and ICU while the processor was in system mode exceeded those issued while the processor was in user mode. Evidently these processes were paging data, and discussions with the users confirmed this suspicion. Re-examination of the workload data, as shown in Figure 5,confirmed that high system intervention occurred on days

displaying below average global performance.
Enforcement of a no-paging data restriction on the compute nodes would require considerable rewriting of the current batch system scheduler. Many of the codes employ automatic arrays whose memory requirements appear only at runtime. There is currently no diagnostic on the SP2 to inform a user of such data paging, short of logging onto the nodes at the time of execution. Users may not be willing to reallocate their scarce real-time resources to repackage their codes to avoid paging.

7. Conclusions

The RS2HPM system reports that the NAS SP2 delivers about 1.3 Gflops daily on a floating-point intensive CFD workload for an overall system efficiency of about 3%. Measurements showed no tendency for this performance to increase over time despite the fact that the NAS SP2 administrators provided an environment for algorithm development. Workload measurements did show a high rate of system overhead for days during which performance was poor and batch measurements showed a similar rate of overhead for poorly performing jobs. The source of this problem appears to be a large amount of data paging induced by node memory oversubscription.That such paging significantly detracted from workload performance was a surprising finding.
The individual batch job measurements indicate that many of the users have not rewritten their codes to take advantage of POWER2 performance features. The ratio of flops to memory references was 1.0, indicating that many of the codes were not making good reuse of the registers. About 50% of the workload floating-point operations resulted from the fma instruction, but the better-performing individual codes perform at least 80% of their operations from fma instructions. Measurements also show relatively high TLB miss rates.
The IBM POWER2 monitor has been quite effective in diagnosing some of the reasons for this performance level. The monitor also allows a confirmation or denial of anecdotal reports of system performance. This tool has identified suboptimal usage of the cache as manifested by high TLB miss rates and a high degree of paging. The ability to monitor the amount of intervention by the operating system was a very useful feature. Other sites wishing to monitor their SP or SP2 systems might consider selecting counter options which could also report I/O wait time in addition to CPU performance.

References

A. Cui and R. Street, “Parallel Computing of Upwelling in a Rotating Stratified Flow”, Proc. 50th Annual Mtg of the Fluid Dynamics Division of the American Physical Society, 1997.
C. Farhat, “Large, Out-of-Core Calculation Runs on the IBM SP2,” NAS News, **2**,11,1995.
Jussi Maki,”POWER2 Hardware Performance Tools”, URL http://www.csc.fi/~jmaki/rs2hpm_paper.
S. Saiyed, et al.”POWER2 CPU-Intensive Workload Performance,” URL http://www.rs6000.ibm.com/resource/technology/SPEC.html
W. Saphir, Alex Woo, and Maurice Yarrow, ”The NAS Parallel Benchmarks 2.1 Results”, NAS Report NAS-96-010, August 1996.
W. Saphir, ”PHPM: Parallel Hardware Performance Monitoring for the IBM-SP2,” NAS Internal Memorandum, October 1996.
C. B. Stunkel, et al,”The SP2 High-Performance Switch”, IBM Systems Journal, **34**, No. 2, 1995.
E.H. Welbon, et al.,”The POWER2 Performance Monitor,” IBM J. Res. Develop.,**38**, No. 5, 545-554, September 1994.
White, S. W. and Dhawan,S.,“POWER2:Next generation of the RISC System/6000 family,” IBM J. Res. Develop.,**38**, No. 5, 493-502, September 1994.
E.Williams and R.Koskela. 1988. Measurement of a scientific workload using the Cray X-MP performance monitor. In Proc. 21st Cray User Group Mtg: 411-422.

Figure 1

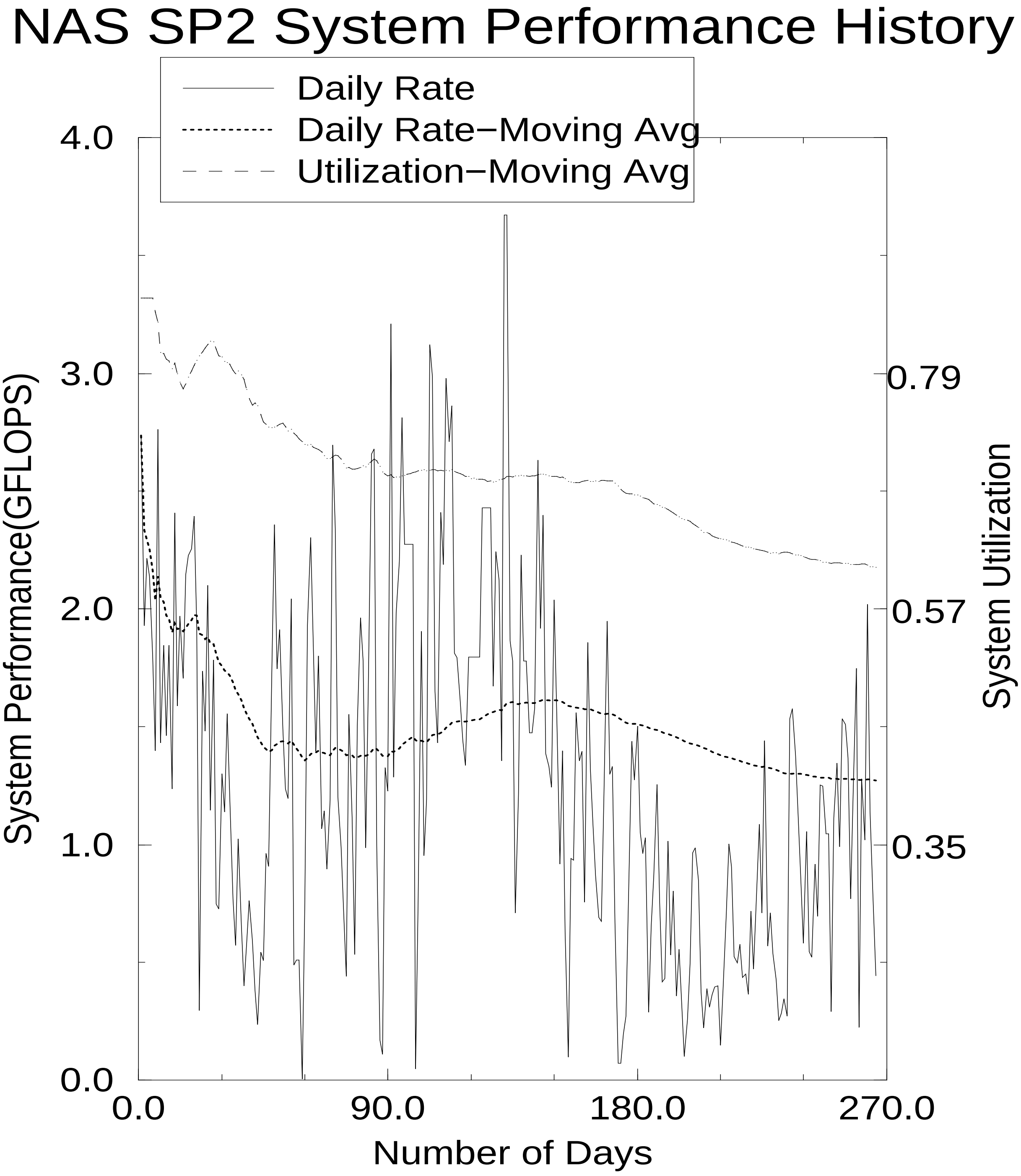

NAS SP2 System Performance History
Daily Rate
Daily Rate−Moving Avg
Utilization−Moving Avg
4.0
3.0
2.0
1.0
0.0
System Performance(GFLOPS)
0.79
0.57
0.35
System Utilization
0.0
90.0
180.0
270.0
Number of Days

Figure 2

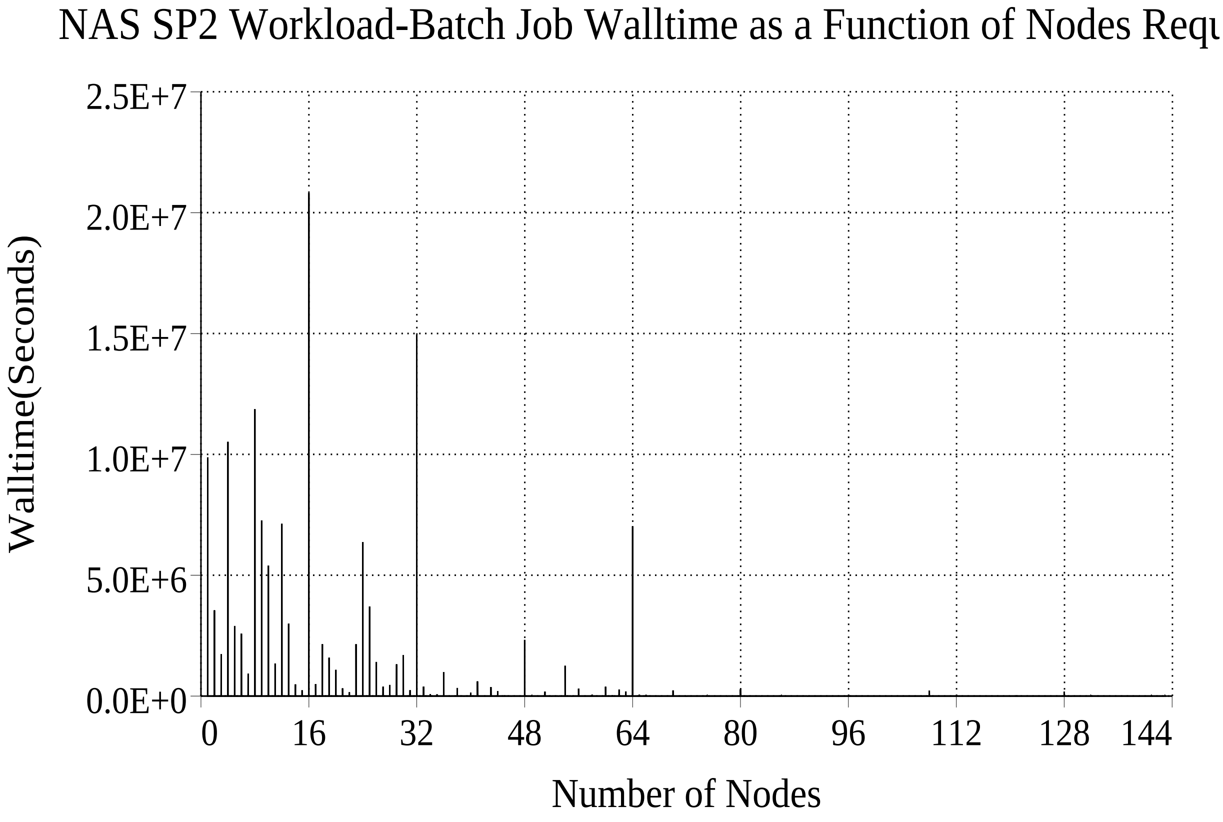

NAS SP2 Workload-Batch Job Walltime as a Function of Nodes Requ
Walltime(Seconds)
2.5E+7
2.0E+7
1.5E+7
1.0E+7
5.0E+6
0.0E+0
0
16
32
48
64
80
96
112
128
144
Number of Nodes

Figure 3

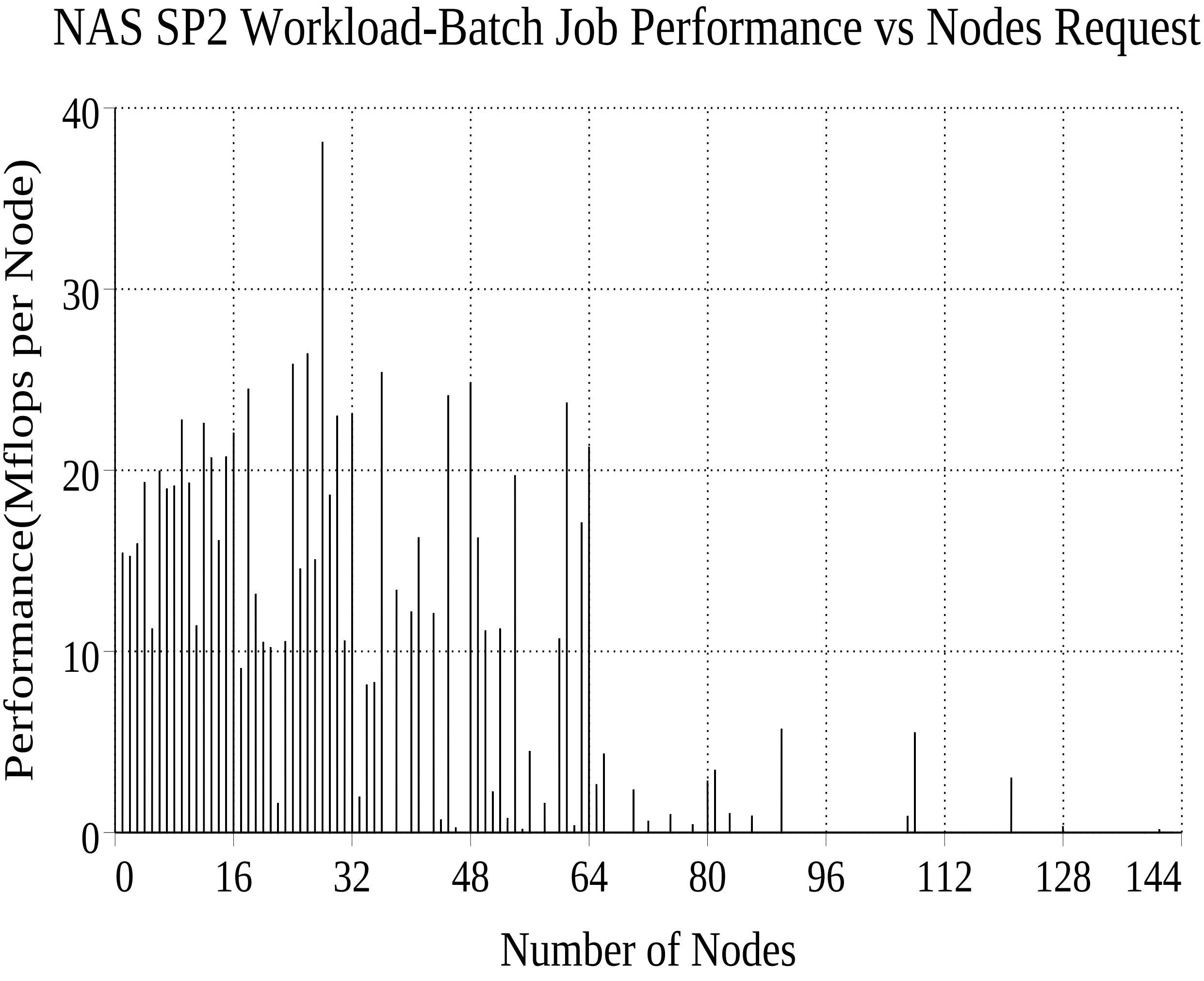

NAS SP2 Workload-Batch Job Performance vs Nodes Request
Performance(Mflops per Node)
0
10
20
30
40
0
16
32
48
64
80
96
112
128
144
Number of Nodes

Figure 4

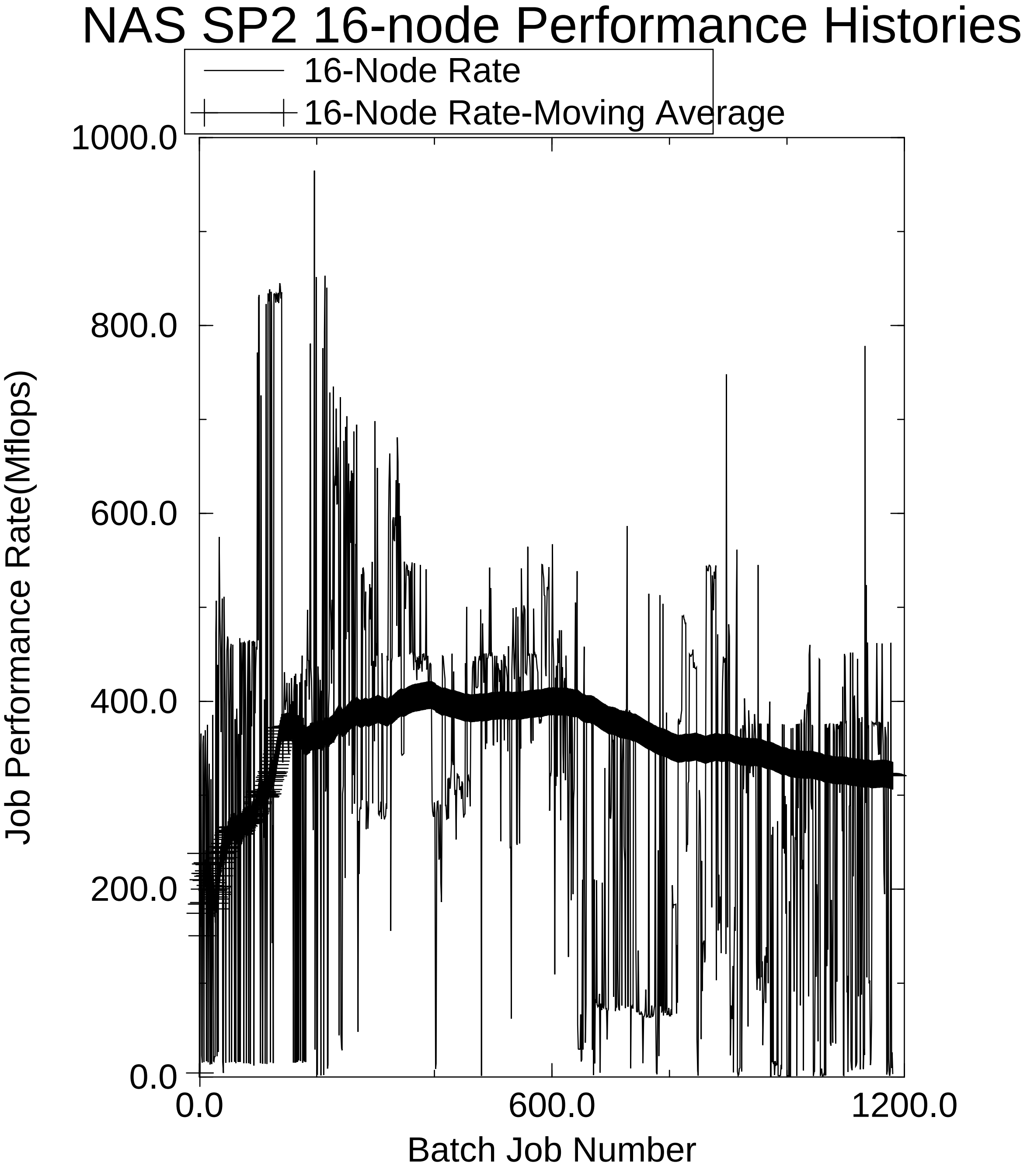
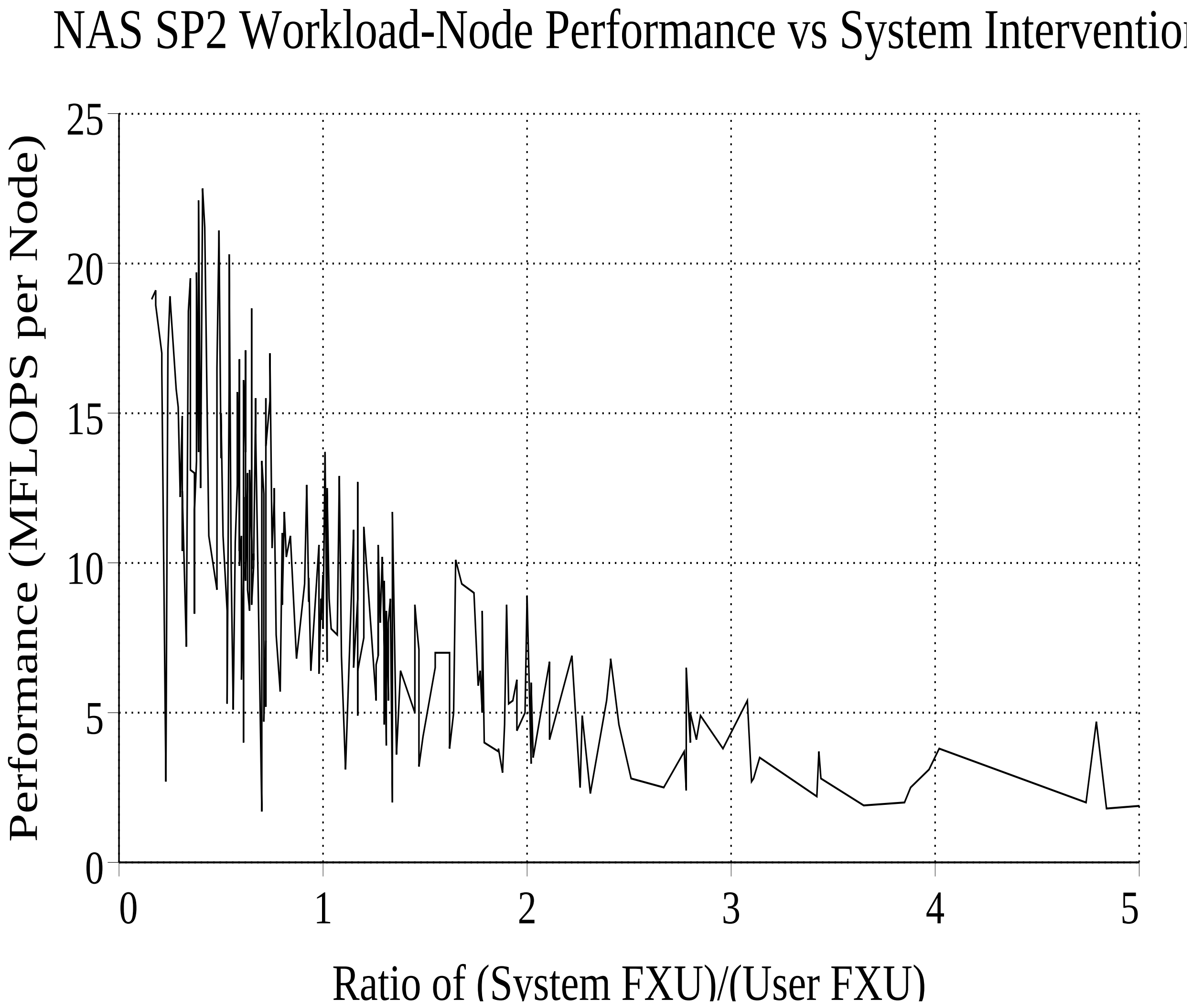

NAS SP2 16-node Performance Histories
16-Node Rate
16-Node Rate-Moving Average
Job Performance Rate(Mflops)
1000.0
800.0
600.0
400.0
200.0
0.0
0.0
600.0
1200.0
Batch Job Number

Figure 5


NAS SP2 Workload-Node Performance vs System Interventio
Performance (MFLOPS per Node)
25
20
15
10
5
0
0
1
2
3
4
5
Ratio of (System FXU)/(User FXU)